Implementations and Applications of Brain-Computer Interfaces

Maxwell J. Marti
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
mart2967@morris.umn.edu

ABSTRACT

Brain-computer interfaces are systems that receive input directly from the brain, and translate it into input usable by a computer. They have been demonstrated to be an increasingly powerful technology that can be applied to many spaces, from disability treatment to entertainment. This paper will discuss interface hardware, data processing techniques, applications of brain input and emerging ideas or areas in the field.

Keywords

Brain-Computer Interfaces, EEG, fNIRS, Classifiers, Active Input, Passive Input

1. INTRODUCTION

For as long as computers have been developed, people have been devising new ways to interact with them. Relatively recently, developments in neuroscience and accompanying hardware have enabled devices to monitor and interpret certain types of brain activity. These devices take many forms, and can be implanted within a person's skull or non-invasively attached to the scalp. When paired with systems designed to interpret the raw data coming from the brain, sensors can become part of a brain-computer interface, henceforth referred to as BCI. These interfaces are unique in that the computer receives input directly from the brain, with no physical movements needed to interact with other intermediaries such as a standard mouse or keyboard. This makes BCIs ideally suited to those who cannot use traditional interfaces, such as those who are paralyzed or have other movement-limiting disabilities.

Figure 1 shows an example of a BCI and its various components. Signal acquisition is performed by the hardware component, which feeds data to a computer for pre-processing and feature extraction. Feature translation is handled by the classifier, which outputs the classified signals to the application's control interface, which then updates the application state as needed.

UMM CSci Senior Seminar Conference, April 2015 Morris, MN.

There are many types of BCIs; this paper primarily covers the operation of non-invasive variants and their applications, as well as selected techniques and algorithms for intermediate data processing. As with any input device, a BCI provides researchers and developers with a new space in which to innovate. Already, BCIs have been used to help people with disabilities regain agency, monitor the mental state of users, and allow new ways to play games and experience entertainment.

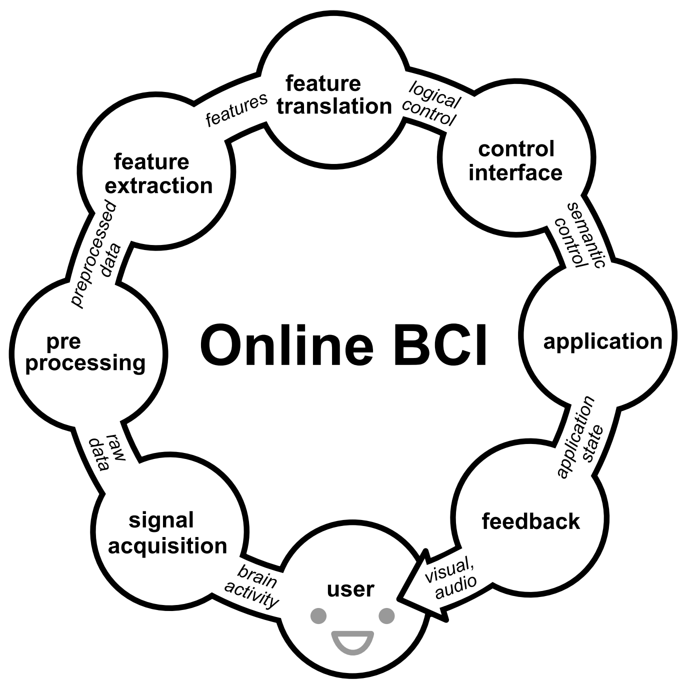

Figure 1: An example of a BCI system, as illustrated by Plass-Oude Bos et al. [9]

2. HARDWARE

2.1 Electroencephalograms

Of the multiple hardware approaches for measuring brain activity, electroencephalograms have been widely used in the development of BCIs. Electroencephalograms, better known as EEGs, are recorded by placing electrodes onto the scalp of a user. The number of electrodes varies between setups, but standards exist that ensure labeling and placement are consistent. The electrodes measure electrical activity in the brain, measuring voltages over time. This process produces a set of waveforms, which can be fed into an input classifier in order to determine what type of action to register.

2.2 fNIRS

Another class of devices used in BCIs use functional near-infrared spectroscopy, or fNIRS to capture signals. This non-invasive method involves measuring the scattering of near-infrared light by red blood cells, with measurements taken through the scalp. The light, emitted by LED or laser diodes, exhibits different patterns of scattering depending on the levels of hemoglobin oxygenation in the blood. These measurements have been shown to be correlated with internal and external stimuli, and are therefore a prime candidate for utilization as part of a BCI [2]. Devices using fNIRS can be used on any region of the scalp, but the prefrontal cortex is the most common measurement location due to its accessibility and importance in higher-level cognition.

3. CLASSIFICATION ALGORITHMS

Once the data has been gathered from the BCI hardware, it must be converted into useful input. The software component responsible for that is called the classifier, which decides what the incoming signals represent. Classifiers consist of algorithms that place different types of input into classes, which typically represent discrete intents by the user. Lotte et al. give a formal definition: "classification consists in finding the true label $y*$ of a feature vector x using a mapping f. This mapping is learned from a training set T. The best mapping $f*$ that has generated the labels is, of course, unknown" [3]. In other words, the designers of a BCI system are responsible for choosing a classification algorithm that correctly labels their input as members of one of some set of classes. Two families of classifiers will be discussed in this section, linear classifiers and neural networks, as well as techniques for improving classifier performance.

When dealing with input from the BCI hardware, system designers must balance several factors. In an EEG system, consideration must be given to which features of the input will be fed into the input classifier. Common features of EEG signals used in classification include amplitude, power spectral density, and autoregressive parameters [3]. Amplitude is a direct measurement of the waveform coming from an EEG and often the most commonly used feature. The amplitude measured from an EEG is around 100 μV, with a frequency of up to 50 Hz. One or more of these features must be extracted, timestamped, and combined to create feature vectors.

Before classifiers are ready to be used with real data, they must first be trained. Classifier training is done by giving the classifier an initial set of pre-classified vectors chosen to mimic real-world input. The classification algorithm must then configure itself to respond correctly to any new data it receives. The training data must be chosen to reduce the risk of overtraining the classifier, which can lead to poor performance when dealing with real data later on. This phenomenon occurs when the classifier is given training data that may not represent real-world data, or is excessively specialized or complex. The classifier will become very good at identifying data that closely matches its training set, but will fare poorly when faced with more natural input and the outliers it contains.

In addition to the classifier training, users must undergo training in order to reliably interact with the BCI. Users practice emitting the correct brain signals, and over time can reduce overall noise and improve the classification accuracy for the BCI. Another facet of user training can be related to training the classifier itself. If the chosen classifier supports machine learning, it can modify its classification output by learning how the user emits brain waves, and tailoring the classification to match them. Other classifiers rely solely on the initial training data, which may be collected from the user and manually classified to produce an accurate representation of that user's input classes. This is important because every human will emit slightly different brain waves, even for the same thought or intent. Care must be taken to minimize the size of the training data set, because the user training process can be lengthy and demanding.

There are several categories that classifiers can belong to. Generative classifiers try to build a statistical model of the classes and their properties. They then compute the probability of the next input being of each class, selecting the most likely; this probability is based on their classification history and the properties of the input. On the other end of the spectrum, discriminative classifiers use training data to build boundaries between classes so that any input can be decisively grouped. The linear classifiers discussed here are of this type, while statistical classifiers such as the Bayes quadratic belong to the generative group. The stability of a classifier refers to the extent to which changes to the training data affect its performance. Linear classifiers have been noted to have a relatively high stability, compared to others such as the MultiLayer Perceptron, a neural network based classifier, which can give more varied results when their training data is tweaked [3]. A desirable property of classifiers is that they are regularized, meaning that they generalize well to real-world data and can account for outliers in their input. To achieve this, the designers of classification algorithms and the researchers who tune and train them must be careful to avoid overtraining.

3.1 Linear Classifiers

Historically, a type of system known as linear classification has been prevalent in BCI research, with two variants composing the majority of the classifiers [1]. The input for a linear classifier are vectors in an n-dimensional space, with the dimensionality depending on the complexity of the signals being captured. The variant known as linear discriminant analysis (LDA) operates by attempting to separate distinct classes using hyperplanes. A hyperplane is a subspace of the containing space that has dimensionality of one less than its containing space; for example, the hyperplanes shown in figures 2 and 3 are one-dimensional lines in a two-dimensional space. Hyperplanes are suitable for linear classifiers because given an n-dimensional space, a hyperplane can always divide it into two subspaces. In LDA, this separation is calculated by maximizing the distance between the target class and the rest of the classes as shown in figure 2; this process is repeated for each class. Because of its relatively low computational cost, LDA is well-suited for use in online systems which must deal with data being streamed in from active BCI hardware [3]. LDA-based systems have had success in many researched applications, but they lose effectiveness when the data becomes less linear, meaning classes may not be able to be adequately defined by linear boundaries.

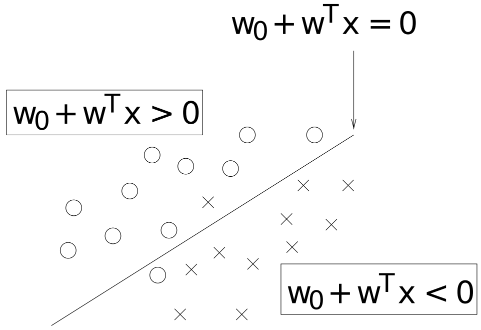

Figure 2: An illustration of LDA separating two classes with a hyperplane, where x is the input vector, w is the weight vector and w_0 is the threshold.

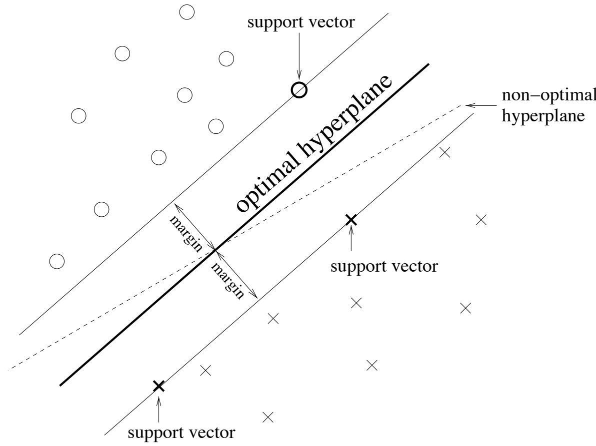

Figure 3: A SVM maximizing the size of the margins between two classes

The second main variant of linear classification is the support vector machine, or SVM. Similar to LDA, linear SVMs use a hyperplane to define class boundaries. The key difference is that the SVM's hyperplane is positioned to maximize the margins between it and the closest points of training data on either side, as seen in figure 3. One advantage of using an SVM is that it has increased capacity for generalization on real-world data. They also mitigate the effects of several issues such as sensitivity to overtraining. SVMs are also resistant to an effect dubbed the "curse of dimensionality", where the "amount of data needed to properly describe the different classes increases exponentially with the dimensionality of the feature vectors" [3]. Parameters can be adjusted to account for the presence of outliers. In addition, SVMs can be adjusted to work in a non-linear fashion by adding a kernel function, which implicitly maps data to higher dimensional spaces [3]. A common kernel function in BCI research is the Radial Basis Function, or RBF:

$$K(x, y) = exp(-\frac{||x - y||^2}{2\sigma^2})$$

Here, x and y are the vector coordinates, and σ is a free parameter. By replacing the feature vectors with a kernel function, an SVM can operate in higher dimensional feature spaces without calculating the coordinates in that space. The function does this by computing the inner products between the images of all pairs of data in the feature space, and its real-valued output depends on the distance from the origin. This has been shown to often be cheaper computationally than direct computation of coordinates in a higher dimension, which is why this method is called the "kernel trick". A SVM that uses the RBF is known as a RBF SVM, or Gaussian SVM.

3.2 Neural Networks

Along with linear classifiers, the other most common type of classifier is the neural network, or NN. At its most basic, a NN is a system that uses a collection of simulated neurons arranged in successive layers that can produce function approximations or decisions based on input data, typically of an unknown format. Each individual neuron's input is attached to the output of the previous layer, and the output layer's neurons determine the class of the input vector. A NN is capable of classifying any number of individual classes. Due to their adaptive nature, neural networks are capable of machine learning, as well as pattern recognition. The most commonly used NN in BCI classification is the MultiLayer Perceptron, or MLP. These classifiers are composed of an input layer, an output layer, and some number of hidden inner layers. As a neural network, MLPs are universal approximators, meaning that they can approximate any continuous function, provided that they are composed of enough neurons and layers [3]. MLPs have been applied to most BCI problem types, and have been proven to be very versatile. However, the adaptive nature of these NNs mean that they tend to be more sensitive to overtraining than linear classifiers, which necessitates a more thoughtful design and selection of parameters.

3.3 Combining Classifiers

While most research focuses on the use of one type of classifier, a recent trend of combining classifiers has emerged. The classifiers used in these combinations and the methods of combining vary, but there are three notable methods. *Boosting* refers to using several classifiers in series, with each one focusing on correcting errors made by the previous ones. These have seen mixed success, with one potential problem being that they are composed of fairly weak individual classifiers which are sensitive to mislabeling [3]. The *voting* method entails feeding the feature vector into a collection of classifiers in parallel, and having the final classification be the class output by the majority. This is the most used of the combination methods, due to its relative simplicity and efficiency. *Stacking* is similar to *voting* in that it uses a collection of classifiers to initially give their own results on one feature vector. These results are then input into a "meta-classifier" that makes the final decision on the classification.

While the success of these methods varies due to the large amount of possible variables, they are often advantageous in that they are likely to outperform a single classifier. The use of different combinations of classifiers in these configurations has yielded promising initial results, and further research may uncover new uses for this model.

3.4 Post-Processing Classified Signals

After the raw input has been processed and classified, there are techniques that can be used to improve accuracy and reduce the effort users must expend. Classified signals may not map directly to computer inputs such as key presses or calls to an application programming interface. The signals can only be assumed to represent different intents on the part of the user. It is therefore necessary for the system to have a post-classification step, in which the classified signals are given new meaning in terms of the target application. This step also represents an opportunity to further process and refine the input. Plass-Oude Bos et al. describe several methods for improving BCI performance after signal classification [9].

The researchers separate their methods into two categories: gathering external evidence and transforming classification. The first category centers around augmenting the data with additional input or contextual inference. It is suggested that by making systems more multimodal, BCI users can expect greater accuracy. This can be achieved, for example, by using additional types of brain signals and events to control the interface, or adding input devices like a keyboard or game controller. This technique provides a greater amount of data for the BCI system to use in its calculations, which can result in improved accuracy if implemented correctly.

Contextual information can also increase accuracy and provide time-saving shortcuts. Consider a simple virtual keyboard application, and the kinds of word-prediction algorithms often found on modern smartphones. By predicting what is most likely to be the next input, the BCI can more accurately choose between two or more input signals that represent different letters. This concept can be extended for whole words and even system actions, given the right implementation. Another method for contextual input is increasing the action space. In contrast to the keyboard example, where the goal was to reduce the number of options and therefore uncertainty, this method relies on a large number of potential actions and a low number of inputs. This makes each input more context-sensitive, such as in a computer game, where players are given the option to “open” any object that supports that action. This approach needs application knowledge, and in some cases may cause signals to become less stationary, meaning that the different contexts will influence how the user perceives their own action, which will in turn be expressed in their emitted signals. An example of this would be akin to a user’s reaction to two different items that have the same context, like coffee and tea; a user may have feelings or associations that introduce noise in their brain signals. The non-stationarity may need to be addressed in order to maintain accuracy, or it could even be used as additional contextual information.

The second category defined by Plass-Oude Bos et al. is transforming classification. These methods apply additional processing to the classified signals. One method is described as a *moving average* algorithm, which multiplies the most recent classifications with a smoothing vector, allowing a more balanced output. This approach works better with input that tends to have a smooth shape over time, for example the throttle of a car, as opposed to the rapid single inputs of a keyboard. Another method in this category is the refractory period, or debounce period. This is when a BCI is prevented from changing states for some period after the last state change, preventing unwanted activations. This method has been successfully applied to applications such as the control of a robotic hand used to grasp objects [9]. BCI designers must balance the need for reduction of unwanted state changes with the desired response time of the system.

4. BCI INPUT PARADIGMS

After the input from the BCI has been classified and any post-processing has been applied, the system must interpret it in terms of the application. This section discusses two paradigms in the use of BCI hardware, each facilitating different kinds of interactions. In both cases, classified signals are converted into data easily consumed by the computer or application programming interface. The key difference lies in how the application is designed to use the data, and the hardware that best meets these needs.

4.1 Active Control

The most direct use of BCI input is to translate it into discrete actions within a computer system. The method used most often in this type of interface involves EEG data. Different interpretations of active control signals can range from binary input to analog mouse-style input [10]. There are multiple signals used in the interpretation of EEG data, one of which is the event-related potential, or ERP. ERPs are the measurable results of the brain reacting to a stimulus or significant event of some kind, which is registered on the EEG. As with many EEG-based interfaces, the participants must undergo training, so that the software can correctly classify the signals it is receiving from the interface. In addition, active control necessitates that a user maintain focus on emitting the correct signals and minimizing mental noise. This reduces the viability of combining EEG with other input methods such as mice or keyboards when using an active control setup, as the brain will be emitting signals in relation to the operation of the other input devices as well as to use the EEG.

There is always the capacity for errors in classification when utilizing an active control interface. Methods to reduce these errors have been proposed, one such method being error-related potentials, or ErrP. A special type of ERPs, ErrPs are the brain signaling that what the interface has done was not what they had intended it to, and can be used as a self-correcting mechanism [4]. In the most simple case of misinterpreting a user’s choice between two options, registering an ErrP can tell the system to immediately rectify the error, and in other situations provide a quick way to reverse the last action taken.

4.2 Passive Input

The concept of passive input has been explored more recently. In this paradigm, instead of utilizing the BCI as the primary input device, data captured from the brain is used as an auxiliary input. Users can interact with other input devices such as standard computer peripherals while wearing the BCI hardware. This has previously been difficult to achieve due to the high amount of concentration that users must maintain in order to reliably utilize a more direct BCI. EEGs, with their greater susceptibility to noise, have been problematic when relied upon for passive input. Often, fNIRS devices have been selected in these applications due to their non-invasive nature and robustness under the effects of noise.

Passive input data has been utilized in multiple applications, with many of them seeking to monitor and improve multitasking activities. The BCI is used to monitor a user’s mental state; it has been shown that fNIRS can distinguish between up to three states of multitasking. The first state is referred to as *branching*, and it occurs when users must keep primary goals in mind while pursuing one or more secondary goals. The second is called *delay task*, and occurs when the user has or is receiving secondary objectives, but decides to delay their pursuit. The final state of multitasking is called *dual task*, and occurs when a user is switching between tasks frequently, without keeping information about non-active tasks in mind [11].

Research on passive input has been done using EEG techniques as well. One such study focused on users controlling a game resembling a simplified version of Pac-Man. At its most intense, the game demanded that users provide directional input via both mouse and EEG simultaneously. Researchers found that participants were comfortable with using both techniques. After performing an analysis of variance (ANOVA) and conducting a questionnaire, the classification of EEG input was found to be accurate, and that the introduction of the mouse input condition did not significantly degrade the performance of the EEG classification [7].

5. APPLICATIONS

Thus far, most applications of BCIs have been focused on restoring agency to severely disabled individuals. As the field has advanced, new applications have emerged that break those boundaries, forming new ideas about what is possible for these interfaces. Many emerging applications rely on the interactive capacity being used for entertainment and games, while others are tying in to technologies that are also undergoing rapid advancement.

5.1 A General User Interface for BCIs

McCullagh et al. describe a system that aims to provide a complete interface to a network of technology [6]. Utilizing the BCI2000 system, which encompasses both EEG and classification in a single package, users can transmit streams of data through UDP, a common lightweight network connection protocol. Their system provides a user access to an XML-based menu, whose items can transmit commands through multiple protocols upon selection. The system is intentionally non-specific, as the goal is to build a framework around which users can interact with all kinds of services, from medical applications to entertainment services, and even control networked smart-home devices. The authors emphasize the difficulty of designing a graphical user interface for BCI users, as the input is limited in complexity. They settle on a four-action menu, with left and right controlling menu traversal, and up and down controlling submenu navigation and item selection, respectively.

5.2 Medical

Often thought of as the main practical use for BCIs, medical interfaces have been developed for as long as the field has existed in any serious capacity. When a patient is unable to communicate or perform important motor functions, a BCI can be used as a work-around. This method of interaction typically requires that the BCI is an active input. Severely paralyzed patients who are unable to speak or type can utilize a BCI such as a speller, a kind of virtual keyboard that responds to input from brain signals. Many spellers have been developed, with several being based on the P300 speller. The P300 is an ERP, utilized here to recognize that a user has selected a letter, and has a latency of approximately 300ms, hence the name. Characters are displayed in a 6 x 6 grid, with the rows and columns flashing alternately and in random order; this is called the Row-Column speller Paradigm (RCP). The user focuses on the desired character and counts the number of times it has appeared. The BCI then determines that the character at the row and column coordinate that registered the strongest on the EEG is the desired character.

While this approach has been successfully employed, there remain issues with accuracy, due to the flashing rows and columns causing the user to become distracted and their focus to waver, with the small distance between two adjacent characters increasing the severity of the problem. Studies have shown that up to 60% of errors are due to this problem [8]. In addition, the visual identification of characters becomes more difficult when they are surrounded by similar items. To address these problems, Obeidat et al. have developed what they call the Zigzag Paradigm (ZP).

In the ZP, every other row of characters is offset to the right by one half the distance between characters on the grid. This maximizes the distance between letters, which reduces the effect of the former issue on selection accuracy. The new spacing of characters also helps alleviate the crowding problem, allowing users to better distinguish individual symbols from their neighbors. In controlled experiments, the ZP achieved 91% accuracy, compared with RCP's 80.6%. Of the ZP-detected errors, 30% were adjacent errors, compared with 43% for RCP. Additionally, ZP correctly detected either the row or column 92.3% of the time, compared with 75% for RCP. Overall, the ZP significantly improved the accuracy of the P300-based speller, as well as reducing user fatigue [8].

Aside from communication, another major interest is the operation of robotics or prosthetics directly from the brain. BCIs have been used to collect motor imagery, or brain signals related to human motor control. These signals can be classified and used as input to devices and machines, allowing users to gain agency in the physical world. BCI-driven machines can take many forms, from prosthetic replacement limbs to full-body robotic exoskeletons, many of which are already in development. The concurrent progress in the prosthetics and robotics fields make these applications some of the most promising, with predictions of how persons with disabilities would be able to regain full movement, as well as general human augmentation.

5.3 Entertainment

A major trend with emerging technologies is to apply them to entertainment applications, especially games. Many researchers use custom or modified versions of existing games to test BCI systems. Often, these modified games help participants in research to better understand the context of their task, such as in the research by Mercier-Ganady et al. using a game resembling Pac-Man. Portraying a task as an interactive game helps to remove subconscious barriers between the user and the interface, especially if the game is immersive. BCI-based games have seen commercial success as well. The 2009 release Star Wars: The Force Trainer used an EEG-like headset to give users control over a "levitating" ball controlled by a variable-speed fan; a sequel planned for 2015 utilizes the same headset and a holographic display.

5.4 Emerging Applications

Many of the techniques currently used to control games using a BCI are being researched for use in new real-world applications. In research done by Poli et al., a BCI is used to control a spacecraft simulator [10]. Participants were asked to pilot a spaceship in two-dimensional space to a predetermined destination while remaining a certain distance from a star. The researchers also combined data from participants in experiments, trying two methods: first, the ERPs were averaged and fed into the classifier, and second, the outputs of the classifier were combined. The first method was able to reduce the noise by approximately 30%, resulting in more accurate classification. One drawback to this method is that an individual's ERPs are unique, and therefore the unique properties may increase classifier uncertainty when combined with others. The second method averaged the output vectors of two participants, and resulted in a noise reduction of the same magnitude as the first, however the researchers note that it does not benefit from the pre-classification noise reduction. Overall, the first method produced results similar to a single user, while the second resulted in a significant improvement in mission success.

Along with new capabilities come potential new concerns, as shown by Martinovic et al. [5]. In their research, they asked users to perform simple image classification. They were able to determine by looking at the BCI data which images were related to or contained sensitive information about participants such as addresses, credit cards, and acquaintances. This introduces the concept of "virtual interrogation", which could be disguised as a mind-training game or outright employed by intelligence agencies. The researchers note that users could counteract this by introducing deliberate mental noise, however this would not help those caught unaware by these potential schemes.

6. CONCLUSIONS

Two types of hardware have been discussed, EEG and fNIRS, showing different methods of detecting input from the brain. There are many types of classification algorithms, ranging from linear models using hyperplanes to neural networks and other systems capable of machine learning. These elements enable the interpretation of brain signals captured directly from the scalp of a user, and together make up a BCI. Medical applications have been developed and continue to advance, with improvements in classification accuracy and user interface design. New applications in the realm of entertainment and games have pushed the technology forward with their broader appeal and need for ever-better features and reliability. Efforts are being undertaken to link BCIs with a large variety of devices through a general purpose user interface. The advances in BCI research have improved the lives of many, and through continued research and development may eventually cause BCIs to become a mainstream element of computing and modern life.

7. ACKNOWLEDGMENTS

Thanks to my professor Elena Machkasova and advisor Kristin Lamberty for their invaluable feedback and support.

8. REFERENCES

[1] O. AlZoubi, I. Koprinska, and R. A. Calvo. Classification of brain-computer interface data. In *Proceedings of the 7th Australasian Data Mining Conference - Volume 87*, AusDM '08, pages 123–131, Darlinghurst, Australia, Australia, 2008. Australian Computer Society, Inc.

[2] M. Ferrari and V. Quaresima. A brief review on the history of human functional near-infrared spectroscopy (fnirs) development and fields of application. *Neuroimage*, 63(2):921–935, 2012.

[3] F. Lotte, M. Congedo, A. Lécuyer, and F. Lamarche. A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4, 2007.

[4] N. V. Manyakov, N. Chumerin, A. Combaz, A. Robben, M. van Vliet, P. A. De Mazière, and M. M. Van Hulle. Brain-computer interface research at katholieke universiteit leuven. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, ISABEL '11, pages 67:1–67:5, New York, NY, USA, 2011. ACM.

[5] I. Martinovic, D. Davies, M. Frank, D. Perito, T. Ros, and D. Song. On the feasibility of side-channel attacks with brain-computer interfaces. In *Proceedings of the 21st USENIX Conference on Security Symposium*, Security'12, pages 34–34, Berkeley, CA, USA, 2012. USENIX Association.

[6] P. J. McCullagh, M. P. Ware, and G. Lightbody. Brain computer interfaces for inclusion. In *Proceedings of the 1st Augmented Human International Conference*, AH '10, pages 6:1–6:8, New York, NY, USA, 2010. ACM.

[7] J. Mercier-Ganady, E. Loup-Escande, L. George, C. Busson, M. Marchal, and A. Lécuyer. Can we use a brain-computer interface and manipulate a mouse at the same time? In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, VRST '13, pages 69–72, New York, NY, USA, 2013. ACM.

[8] Q. Obeidat, T. Campbell, and J. Kong. The zigzag paradigm: A new p300-based brain computer interface. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 205–212, New York, NY, USA, 2013. ACM.

[9] D. Plass-Oude Bos, H. Gürkök, B. Reuderink, and M. Poel. Improving bci performance after classification. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, pages 587–594, New York, NY, USA, 2012. ACM.

[10] R. Poli, C. Cinel, A. Matran-Fernandez, F. Sepulveda, and A. Stoica. Towards cooperative brain-computer interfaces for space navigation. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 149–160, New York, NY, USA, 2013. ACM.

[11] E. Solovey, P. Schermerhorn, M. Scheutz, A. Sassaroli, S. Fantini, and R. Jacob. Brainput: Enhancing interactive systems with streaming fnirs brain input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2193–2202, New York, NY, USA, 2012. ACM.